Array Variables

Version 8

January 29, 2024

Developed by TRP Laboratories

Version History

Version	Date	Revision Description
1	1/15/2014	Initial publication
2	8/23/2018	Format changes
3	3/25/2020	Retrofit to new template and publish
4	8/2/2021	Added hyperlinked cross-references to usage help
5	12/20/2021	Added content describing units modification in *Section 3 Array Variable Initialization* on page 3 Added *Section 3.1 ARRAY_UNITS* on page 4 Revised cyflex.com usage help *for* `sarr` to add supporting content for strings and computed expressions
6	5/12/2022	Updated all hypertext cross-references to cyflex.com usage help descriptions.
7	6/2/2022	Revised *Section 2 Array Variable Specifications* on page 2 to add mention that up to 3500 `enums` may be used in an array.
8	1/29/2024	Rebrand to TRP Laboratories

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.

  Example: Select the `cmdapp-`*`relVersion-buildVersion`*`.zip` file….

- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.

  Example: **Type**: Click **Select Type** to display drop-down menu options.

- Cross-references are designated in Arial italics.

  Example: Refer to *Figure 1*…

- Click intra-document cross-references and page references to display the stated destination.

  Example: Refer to *Section 1 Introduction on page 1.*

  The clickable cross-references in the preceding example are *1*, *Introduction*, and on page 1.

Related Document

CyFlex® Variables, Units and Computed Expressions

CyFlex Documentation

CyFlex manuals are available at https://cyflex.com/. View **Help & Docs** topics or use the **Search** facility to find topics of interest.

Table of Contents

1 Introduction

Array variables provide a mechanism that allows closely related parameters to be referenced by a common name, for example exhaust port temperature for each cylinder. The types of array variables supported are:

- REAL_ARRAY_VARIABLE
- INTEGER_ARRAY_VARIABLE
- LOGICAL_ARRAY_VARIABLE
- STRING_ARRAY_VARIABLE

Specify the maximum number of array variables allowed in a running system as an argument to the process `sys_start`. This process is normally spawned very early in the `go` script. Refer to cyflex.com usage help for `sys_start` for related information.

If the number of array variables has not been specified and array variables are created via the translator `arr_specs`, the maximum number of array variables will be predicted. This is not recommended as problems usually result when additional array variables are created later. Refer to cyflex.com usage help for `arr_specs` for related information.

2 Array Variable Specifications

Array variables are created via a specification file. The `arr_specs` translator reads the file and then creates the array variables defined in the file. Each array variable specification must consist of a minimum of three lines.

The first line contains:

- variable name
- variable type
- units
- display resolution

The second line contains the number of elements making up the first dimension of the array. This line may also contain strings called `enums` that describe each element of the dimension. An `enum` string must not start with a number. These strings are optional and if no strings are entered, each element of the dimension will be known as `0, 1, 2 … n`. Up to 3500 `enums` may be used in an array.

If the array has additional dimensions, the 'second line' is repeated until all array dimensions have been specified. When all elements of the array have been defined, they are followed by a line that consists of a `$`. This indicates the termination of the array variable definition.

The preceding format can be repeated for any other array variables that need to be defined.

The following is a snippet of an array variable specification for emission concentrations.

```
#Variable name      Variable type    units    display resolution
Conc                 REAL_ARRAY       ppm              2
#dimension   associated
#   size       enums
     7       - CO CO2 LCO O2 NO THC
$
```

The preceding specification defines a real array variable `Conc` with units of `ppm` It has seven elements and the display resolution is two decimal points. Each of the elements will be known as `0,1 … 6 or CO, CO2 … THC.` The dash (-) `enum` has a special meaning. When a dash is specified as an `enum`, it indicates that this element should not be displayed by the utility `garr` even though the element exists. Refer to *Section 4 Array Variable Utilities* on page 5 for more information on `garr`.

3 Array Variable Initialization

The default value of an array variable is zero or NULL in the case of string variables. This default value may be overridden by specifying initialization values in the specification file. The initialization values are specified after the lines that define the array variable. There may be multiple initialization lines and they are executed in the order that they appear in the file. The syntax of an initialization line is:

`Variable_name:[start_element][,end_element]…=value_start[,increment]`

Where:

- *`Variable_name`* is the name of the array variable being initialized with a colon, ':', appended to the name. [Required]
- `start_element` is the number or `enum` of the array element of the first dimension where initialization will start. [Optional]
- `end_element` is the number or `enum` of the array element of the first dimension where initialization will stop. The value is prepended with a comma, ','. [Optional]
- … repeats the above for any additional dimensions. [Optional]
- `value_start` is the first initialization value. The value is prepended with an equal sign, '='. String array values are initialized with the contents of whatever follows the equal sign. The string may contain blanks and does not need to be enclosed by quotes. [Required]
- `increment` is the incremental value to be added to `value_start` for each subsequent array element initialized. The value is prepended with a comma, ','. The `increment` is not used for string variable initialization. [Optional]

The preceding array could be initialized as in the following example:

```
#Variable name      Variable type    units    display resolution
Conc                 REAL_ARRAY       ppm              2
#dimension   associated
#   size       enums
     7    - CO CO2 LCO O2 NO THC
     Conc:=-999.
     Conc:CO2,NO=1.2,.5
$
```

The first initialization line sets every element of the array variable `Conc` to `-999`.

The second initialization entry would then set the `CO2` element.

The third element would be set to `1.2`.

The fourth element, `LCO`, would be set to `1.7`.

The fifth element, `O2`, would be set to `2.2`.

The sixth element, `NO`, would have a value of `2.7.`

All other elements of the variable would have a value of `-999`.

String array variable initialization example:

```
#Variable name      Variable type    units    display resolution
Range               STRING_ARRAY     -            -
#dimension   associated
#   size       enums
     5    CAI_HLD  CAI_THC CAI_CH4  FTIR_FID  FTIR_PMD
     Range:=CAI
     Range:CAI_HLD=this is HLD
     Range:FTIR_FID=this is FID
$

“Range:CAI”  -- initializes all 5 members to ‘CAI’
“Range:CAI_HLD:=this is HLD” – sets Range:0  to ‘this is HLD’
“Range:FTIR_FID=this is FID”  -- sets  Range:3  to ‘this is FID’
Range:1 contains ‘CAI’
Range:2 contains ‘CAI’
```

3.1 ARRAY_UNITS

The units of REAL_ARRAY and INTEGER_ARRAY variables are initially set to the units listed on the first line of the specification. The units of individual members (or all members) may be modified by specifying the alternative units along with the value.

```
#Variable name      Variable type    units    display resolution
Conc                 REAL_ARRAY       ppm             2
#dimension   associated
#   size       enums
     7    - CO CO2 LCO O2 NO THC
     Conc:=-999.[%_conc]
     Conc:CO2,NO=1.2,.5
$
```

The first initialization line sets every element of the array variable Conc to -999. [%_conc]

The second initialization entry would then set the CO2 element to 1.2[ppm]

The fourth element, LCO, would be set to 1.7[ppm].

The fifth element, O2, would be set to 2.2[ppm].

The sixth element, NO, would have a value of 2.7[ppm].

All other elements of the variable would have a value of -999.[%_conc]

4 Array Variable Utilities

Several commands can be used to access array variables. They obtain the current value of an array element or change the value of an array element.

4.1 gvar

Use the `gvar` command to obtain and display the value of a variable. The syntax is slightly different since you are accessing an array variable. For example, enter the following to obtain the value of the `CO2` element of the above array variable:

```
gvar Conc:CO2
```

The above could also be entered as:

```
gvar Conc:2
```

Both of the above commands would display the value of the third element of the array variable `Conc`.

An element of an array variable may also be obtained by setting another variable to the value of the element you wish to display. Enter the following:

```
gvar Conc?elem_num
```

The above command first determines the value of the variable `elem_num`. When the value is known, that element of the array variable is obtained. Therefore, if `elem_num` has a value of two, the second element of the array variable `Conc` would be displayed. The same element would be displayed if `elem_num` contained `CO2`. The variable `elem_num` must be either an integer variable or a string variable.

Refer to cyflex.com usage help for `gvar` for related information.

4.2 svar

Use the `svar` comment to change the value of an array variable element. The syntax for changing array variables is the same as described above with one additional entry. The name of the array variable element is followed with the value that it should be set to. For example,

```
svar Conc:2 12.3
```

The above sets the third element of the variable `Conc` to a value of `12.3`. All of the various forms of the specified ‘element’ described above may be used for this command also.

Refer to cyflex.com usage help for `svar` for related information.

4.3 garr

The `garr` command displays array variables only. The following is the command usage:

```
garr [arr_vrbl]:[enum_str]:[enum_str2] ...
```

or

```
garr [arr_vrbl]?[int_var]?[int_var2] ...
```

Refer to cyflex.com usage help for `garr` for related information.

4.4 sarr

Use the `sarr` command to set an array variable with a value. The following is the command usage:

Refer to cyflex.com usage help for `sarr` for related information.

4.5 @set_array()

In addition to the command `sarr`, array variables may be set via computed expressions. The function `@set_array()` has been added to the valid computed expressions that will provide this capability. The arguments to `@set_array()` are the same as the arguments to `sarr`. The only difference is that the arguments must be enclosed with single quotation marks.

For example:

"`set_array( 'my_vrbl:TC15', '1234' )"`

When the above computed expression was evaluated, all elements associated with the `enum` `TC15` of the array variable '*`my_vrbl`*' would be set `'1234'`.